White Paper Report

Report ID: 98513

Application Number: HD5079409

Project Director: James Allan (allan@cs.umass.edu)

Institution: University of Massachusetts, Amherst

Reporting Period: 9/1/2009-8/31/2010

Report Due: 11/30/2010

Date Submitted: 12/20/2010

Final Performance Report

HD5079409

Digital Humanities: Level II

OCRonym

Entity Extraction and Retrieval for Scanned Books

James Allan and David A. Smith
University of Massachusetts
Amherst, MA

30 November 2010

Our goals in the OCRonym project involve building better language models to aid in correcting OCR transcription of proper names and disambiguating names to improve access to large scanned book collections. In our work on this project, we have made progress in:

- scaling up language models that automatically adapt to the vocabulary of different authors and genres (§1);
- speeding up syntactic analysis of text to aid in named-entity recognition (§2);
- clustering terms within books (§ 3) and within dynamically retrieved collections of similar passages (§4); and
- building a prototype system that allows users to combine full-text search with faceted browsing of names that occur in the context of search terms (§5).

As we describe in §3, we found that one of our original hypotheses was wrong: due to relatively simple language models used in OCR, transcription errors do *not* cluster more frequently in named entities than in the general vocabulary. (Paradoxically, therefore, many language-model-based methods of OCR error correction could hurt name accuracy while improving overall transcription.) It reamins true that names are an important use case for searching digital collections and that the accuracy of name *detection* in the Internet Archive's scanned book collection is substantially below that of the news-domain training data.

We are therefore continuing work on this project past the lifetime of this grant in the two areas of bootstrapping name classifiers (§4) and improving name search (§5). This work is important to our ongoing collaborations with Tufts University's Perseus Digital Library Project and to the Internet Archive, whose texts form our testbed.

1 Topic Modeling from Noisy Documents

Optical character recognition (OCR) is one of the great success stories of computer vision and machine learning. Widely-available commercial systems typically achieve over 99% character accuracy for high-quality images of modern documents with reasonably simple layouts. However, large-scale efforts to digitize printed sources back to Gutenberg have highlighted regions of much lower performance: in the Internet Archives collection of over one million public-domain scanned books, sample evaluations reveal character accuracies of around 90% even on English-language texts, with higher error rates on non-English Roman texts.

While these error rates are adequate for many retrieval tasks, in which users are simply presented with the appropriate page image, other "downstream" applications of NLP—from information extraction to machine translation—will suer much more from text where it is expected that one in every ten characters is in error.

Informally, the process that transcribes words into marks on a page can be viewed as a noisy channel. OCR is therefore the corresponding inverse task of inferring the transcription from this channels output, possibly with the assistance of a source model of the underlying text. Any model used to perform OCR is known as a channel model. While most commercial OCR systems use very little source (language) modeling, the success of language modeling in remedying deficiencies in inaccurate channel models in other tasks, such as speech recognition and machine translation, has suggested their use in OCR correction to previous researchers. However, while those other tasks usually employ language models to rescore a lattice or n-best list of alternatives, most of the top-performing OCR systems are commercial black boxes, so researchers only have access to their "one-best" output. Previous approaches that utilize the one-best

output of commercial systems have performed traditional rescoring by constructing a channel model that hallucinates a lattice or n-best list, which is then rescored by a language model (Tong and Evans, 1996; Kolak et al., 2003).

While the large amounts of generic electronic text available for language modeling have resulted in impressive results in machine translation and other tasks of late, the wide variety of topics, periods, genres, and styles in the Internet Archives book collection suggests that the corpus itself, albeit noisy, should instead be used to train language models for OCR correction.

For this project, we are therefore investigating a novel, topic-based language modeling approach to OCR correction that aims to model the vocabulary variation in a large, diverse corpora, such as the Internet Archives book collection, while still capturing the local dependencies necessary for accurate word prediction. We outline two approximate decoding techniques to perform OCR correction without the need for an explicit channel model (Naradowsky et al., 2010).

We assume as input the one-best OCR transcriptions made by the Internet Archive with the ABYY FineReader system. For simplicity and computational efficiency, we restrict ourselves to books in English.

Statistical n-gram language models (see Chen and Goodman, 1998, for an overview) decompose the probability of a string of text (e.g., a document) into a product of probabilities of individual words given some number of previous words. Equivalently, these models assume that documents are generated by drawing each word from a probability distribution specic to the context consisting of the immediately preceding words. One aw in this method is that word usage can be highly topic-dependent. For instance, "I'll be in the—" is likely to be followed the word pub in an email about weekend plans. In an email about a business meeting, however, the word "office" is clearly more likely.

In contrast, probabilistic topic models, such as latent Dirichlet allocation (Blei et al., 2003), model documents as nite mixtures of specialized distributions over words (known as topics). These models assume that documents are generated by choosing a document-specic distribution over topics, then repeatedly selecting a topic from this distribution and drawing a word from the topic selected. Word order is ignoreddocuments is modeled as a "bags-of-words".

For large, heterogeneous document collections, such as the Internet Archives book collection, it is especially important to take topics into accountany single model of language is likely to be a poor model of the text in any particular book. In this paper, we therefore aim to leverage both word order and topic information for OCR correction by using a topic-based language model (Wallach, 2006). A bigram version of this model has been shown to outperform even a trigram language model without topics (Wallach, 2008).

For evaluation purposes, we identied a subset of the scanned books that have also been transcribed by Project Gutenberg. We produce a forced alignment between the Project Gutenberg and the OCR transcriptions to calculate the error rate. The "documents", for purposes of the topic-based language model, are paragraphs, both in training and testing contexts. Since the books in the collection average 200 pages, book-level topics would be far too diffuse.

We experiment with two dierent decoding/correction strategies for the topic-based language model. For the rst, we perform "left-to-right" inference for each OCR transcript (Wallach et al., 2009). At each position, we x the previous words in the document and infer the latent topic assignments by observing only the prex of the document up to the current position.

We then extract the language model's prediction of the next word and compare it to the probability of the word actually observed in the transcript. If the predicted word is more probable than the observed word by some threshold τ, we replace the observed word. In subsequent processing of the transcript, we use this newly predicted word in lieu of the observed word.

The second decoding/correction strategy evaluates the language model at each position independently.

This technique assumes that the errors in the baseline OCR system are not highly correlated. If the baseline system had itself used an n-gram language model, this assumption would be much less tenable, but, apart from errors arising from a deciency in the underlying image, it is realistic to treat OCR errors in turn as independent. We therefore treat each word position as unobserved while treating all the other words in the documents as observed. We predict the word at the current position as before and replace the word if the probability of the new prediction outstrips that of the true observed word by a threshold τ. These point-wise decoding procedures can be performed in parallel.

We are currently performing an empirical comparison of these two decoding strategies using Wallach's topic-based language model (2006).

2 Fast, Eager Dependency Parsing

The topic-based language models discussed above learn the correlation patterns of words—i.e., they learn specialized distributions of words for each page to improve OCR correction. Even with these specialized distributions, however, these language models still assign a probability to each word that depends only on the previous one or two words and the inferred "topic".

In order to capture longer-distance dependencies among words, we are working on efficient methods for syntactic dependency analysis. To date, the application of syntactic methods to language modeling for large-scale corpora has been hindered by their inefficiency compared to "count-and-normalize" n-gram methods. While some syntax-based language models may surpass n-gram models trained on a comparable amount of data, the n-gram models can quickly ingest much larger training sets. Our research, which is currently under review (Wu and Smith, 2010), aims to reduce this gap in efficiency.

Current research in data-driven dependency parsing can be separated into *graph-based* and *transition-based* methods (Nivre and McDonald, 2008). Graph-based methods view dependency parsing a sentence as a structured prediction problem whose output is a single (labeled) directed graph. The features of this structured prediction model encode constraints about which edges, pairs, of edges, etc., should appear in this graph. If these constraints apply only to single edges, $O(n^3)$ algorithms optimally solve this graph-prediction problem for both projective and non-projective trees. Higher-order constraints can improve empirical accuracy and linguistic plausibility—at the cost of making optimal projective parsing slower and non-projective parsing intractable (McDonald and Satta, 2007). Many state-of-the-art graph-based methods thus turn to approximate inference techniques (McDonald and Pereira, 2006; Smith and Eisner, 2008; Martins et al., 2009).

Transition-based dependency parsers, in contrast, aim to select the a sequence of appropriate actions to take during a tree construction process (Nivre, 2008). This tree-construction often proceeds from the beginning to the end of the input sentence (incrementally), which more plausibly models human sentence processing. Transition-based parsers have the additional advantage that parsing time for projective trees is linearly dependent on the length of sentence (quadratic for non-projective). Among transition based parsers, stack-based shift-reduce parsers have shown state-of-art performance by making shift and attachment decisions based on local features.

There is, however, a fundamental asymmetry in these stack-based incremental algorithms. When a token is a left branch child of its parent, meaning it appears before its dependency parent in the sentence, the parser does not have any knowledge of its parent and must make a shift decision based on the absence of certain features on the stack. On the other hand, when the parent precedes the child token, the parser makes an attachment decision based on the presence of certain features. In a sense, the shift decision on a left branching child delays its attachment until some expected features show up. Successful shift-reduce

training size	1 sec. xfm.	1 sec. rev.	4 sec. xfm.	4 sec. rev.	20 sec. xfm.	20 sec. rev.
MaltParser	N/A	67 (73,74)	N/A	73 (78,79)	N/A	79 (83,84)
RB Parser	74 (84,79)	71 (77,78)	77 (86,82)	74 (80,80)	82 (88,85)	79 (82,85)

Table 1: Comparison between MaltParser and the new RB parser. The evaluation shows labeled attachment scores, unlabeled attachment scores, and label accuracies, in that order. The RB parser can learn more quickly from smaller amounts of training data.

parsers often use look-ahead methods, such as reading more tokens from the input buffer, to get around this asymmetry. Nevertheless, the number of tokens to be considered is often arbitrary and perhaps inadequate to deal with longer span left arcs.

The asymmetry between left and right children in a dependency tree often requires a shift-reduce parser to delay an attachment decision until all relevant tokens are visible to the parser. We can eliminate the need to delay decisions by transforming all dependency trees into right branching trees. In right branching trees, all children must appear after the parent token in the sentence. Therefore each token must always make an attachment when it shows up on top of the input buffer, without delay.

The performance of the new *RB parser*, based on this right-branching transformation, closely matches that of the state-of-the-art MaltParser (table 1). The labeled attachment score is 78.53%, insignificantly lower than MaltParser. Unlabeled attachment score becomes 82.42; label accuracy becomes 84.68, higher than MaltParser's 83.93. A further point of interest is that the RB parser achieves higher accuracies on smaller amounts of training data.

Although on the full training set the transformation does not significantly alter parsing accuracy, it has a large advantage in parsing time. On a 2.1GHz processor, we ran the RB parser and MaltParser on all 24 sections of the Penn Treebank WSJ corpus. The RB parser spent 75.5 seconds to read and parse all 49,208 sentences, of which just under 64 seconds are spent on parsing. The transformation and reversal of all sentences took 24 seconds each. The sum of these is still smaller than the MaltParser by a factor of ten, which parsed the whole set in 996 seconds. Although both the MaltParser and the RB parser use the logistic regression learner in LIBLINEAR (Fan et al., 2008), the light weight of our RB parser implementation may have contributed to this speed-up. More importantly, the transformation on average lowers the number of possible actions given each parsing state, since left-attach actions are no longer used. Upon close examination, we discovered the set of possible actions is particularly small when the parsing is considering transformed arcs.

3 Clustering for OCR Correction

One of the working hypotheses of this project was that names, which are more likely to fall outside dictionaries and language models used by OCR systems, were more likely to contain errors that other terms were. Despite the evidence of some previous work, however, we found that this was not the case (table 2): names did not differ significantly from non-names in their character error rates. Admittedly, these results were measured on books with fairly clean OCR—all character error rates were less than 4% when punctuation was removed. Texts with higher error rates, in any case, would display much less accurate NER performance for performing this evaluation.

This similarity in error rates suggested two lines of attack on the name recognition problem. First, since it seemed that the out-of-vocabulary rate for names and non-names was more similar than we had estimated, we extended our cluster-based language modeling approach to the whole vocabular rather than names alone.

	Character error rate across six books					
Percent name errors	1.65%	2.02%	0.26%	1.58%	1.28%	0.65%
Percent overall errors	1.02%	3.81%	0.30%	0.98%	1.20%	1.29%

Table 2: The percentage of incorrect characters in names is not significantly greater than the percentage of incorrect characters overall. The error rates do not include punctuation.

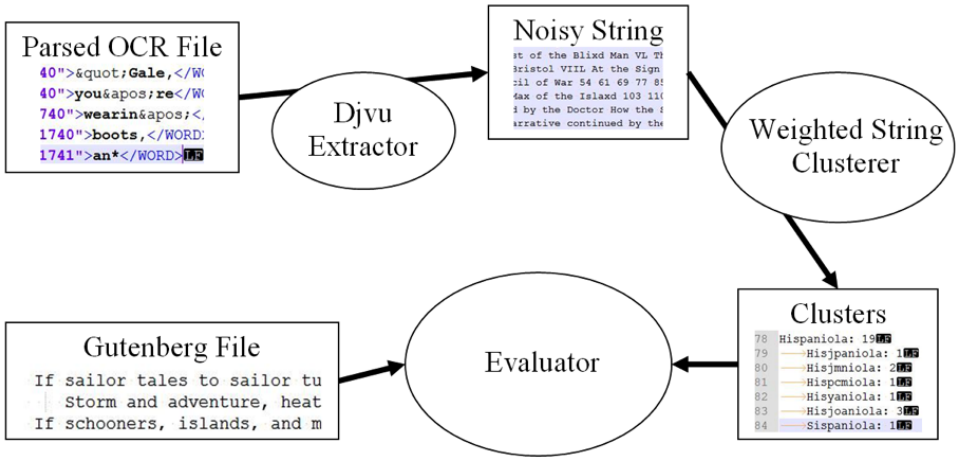

Figure 1: Clustering possible errors

We descrbe the clustering approach in the rest of this section. Second, during the evaluation, we discovered that many of the errors in name recognition did not come from OCR errors but from the NER system not recognizing names in new contexts. We therefore pursued a boostrapping strategy to retrain NER systems on cross-document clusters of similar contexts (§4).

Figure 1 shows the outline of our clustering approach. The raw OCR text is stripped of its markup and tokenized on whitespace and punctuation. We then performed greedy agglomerative (bottom-up) clustering on the vocabulary of the document using a weighted edit-distance measure. At each iteration, we find the pair of clusters with the lowest edit distance below our threshold and merge them. Words with fewer than five characters are not included in the clustering process.

A further restriction on cluster merging uses the notion of a *headword*: we hypothesize that an incorrectly transcribed term will be reliably correctable when it is mostly transcribed correctly elsewhere in the document. Each cluster therefore distinguishes its most frequent variant transcription as its headword; two clusters can only be merged if one cluster's headword frequency is at least ten more than the other's. When no clusters can be merged, the process stops.

Given a clustering of the terms in a document, we then replace all terms in a cluster with the headword. We evaluate the accuracy of the correction process using OCR'd books that have also been transcribed by Project Gutenberg. The Gutenberg and OCR transcripts are aligned with unweighted edit (Levenshtein) distance, and we can then calculate character and word error rates. The choice of edit distance measure and merging threshold in the clustering process has an important effect. While unweighted edit distance permits acceptible performance, it is more effective to train the edit distance model on held-out data: when OCR transcripts and Gutenberg books are aligned, we can learn a model of likely substitutions, such as $e \rightarrow c$ or $i \rightarrow l$, or contextual insertions, such as $h \rightarrow li$.

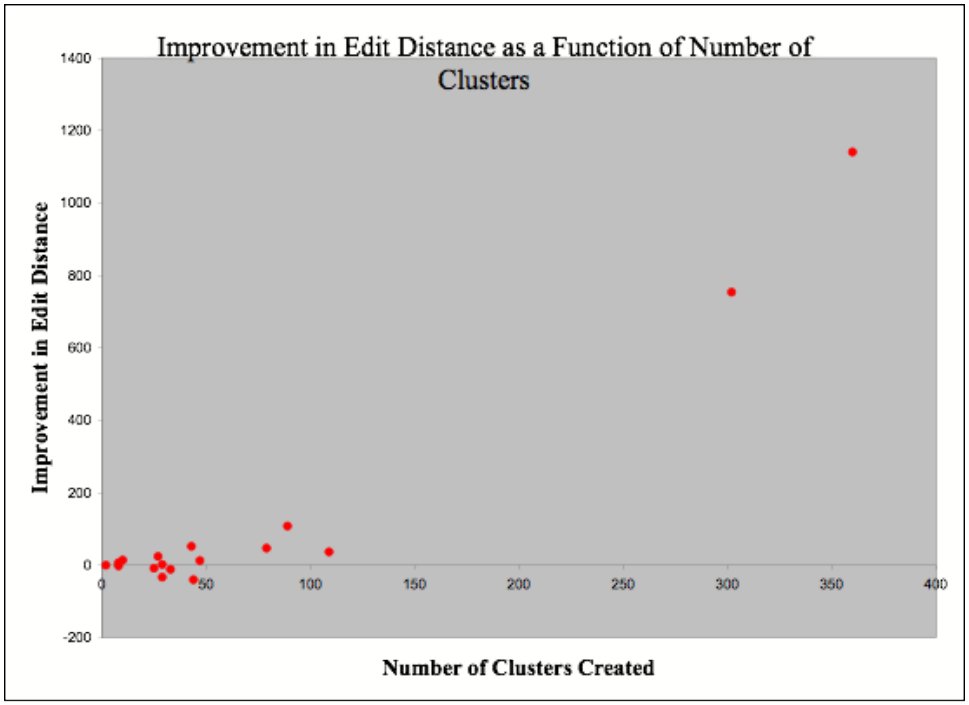

Figure 2: Correlationn between the number of clusters created and OCR improvement is 0.96. When two outlying books with large numbers of clusters and corrections are removed, the correlation is still 0.60.

The size of a document and the distribution of terms within it will obviously affect how well this clustering process works. In our evaluation set, there is a wide range of clusters produced by our process for each document—from 2 to 360, although most books have under 100 clusters (figure 2). There is a very strong correlation, even without outliers, between the number of clusters produced and the effectiveness of the OCR correction algorithm. The greater effectiveness on larger books leads us to believe that stopping the clustering process at book boundaries could be costing us some performance. It would therefore be interesting in future work to explore the tradeoffs between clustering larger datasets together and exploiting the bursty distributions of content words, including names, in individual books. In the next section, we discuss preliminary work on one such cross-book approach.

4 Bootstrapping Name Classifiers

While it may still be worthwhile to cluster the entire collection of two million scanned books from the Internet Archive, such an undertaking is computationally expensive. We have therefore started to perform experiments on using document retrieval techniques that, in effect, perform the clustering on-line when the user is focused on a particular document. Given a passage (usually a paragraph) with candidate names in it, find similar passages in the collection using all terms in the passage. We can then cluster terms and retrain our named-entity recognition system within this set of similar documents.[1] If the clusters are large enough, we can simply use the plurality output of the NER system on a name in the cluster (Finkel et al., 2005). It is not, however, ideal for the set of similar passages to be *too* similar: if all the retrieved passages were

[1]That is to say, the brute force complexity is linear in the (large) size of the collection, rather than quadratic.

duplicates, there would be no variation in context for the NER training to exploit.

This work is continuing beyond the term of the grant. Importantly, the work on this task will include features generated by all of the separate efforts in language modeling, parsing, and clustering so far.

5 Faceted Searching with Names

The ultimate goal of our work on language models for OCR correction and named-entity recognition is to improve access to the scanned book collection. We have created a prototype system that combines full-text search with name browsing. When, for example, the user performs a search for "jolly roger", he may find several pages, some of them, from *Treasure Island*, associated with the personal name "Ben Gunn" and "Jim Hawkins" and the place name "Skeleton Island". The user might search for "commodities" in works on economics, and restrict the results to those pages that contain "Portugal" or "Spain". In addition to delivering the eventual results of our research to end users, this search interface will prove useful during development by making misclassified names and recognition errors more apparent.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pages 363–370, 2005.

Okan Kolak, William Byrne, and Philip Resnik. A generative probabilistic OCR model for NLP applications. In *Proc. of NAACL*, pages 55–62, 2003.

Andre Martins, Noah Smith, and Eric Xing. Concise integer linear programming formulations for dependency parsing. In *ACL*, pages 342–350, 2009.

Ryan McDonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *EACL*, 2006.

Ryan McDonald and Giorgio Satta. On the complexity of non-projective data-driven dependency parsing. In *IWPT*, pages 121–132, 2007.

Jason Naradowsky, Hanna Wallach, and David A. Smith. Topic modeling from noisy documents for OCR correction. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, 2010.

Joakim Nivre. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553, 2008.

Joakim Nivre and Ryan McDonald. Integrating graph-based and transition-based dependency parsers. In *ACL*, pages 950–958, 2008.

David A. Smith and Jason Eisner. Dependency parsing by belief propagation. In *EMNLP*, pages 145–156, 2008.

Xiang Tong and David A. Evans. A statistical approach to automatic OCR error correction in context. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 88–100, 1996.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984, 2006.

Hanna M. Wallach. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Xiaoye Wu and David A. Smith. Right-branching tree transformation for eager dependency parsing. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, 2010.